

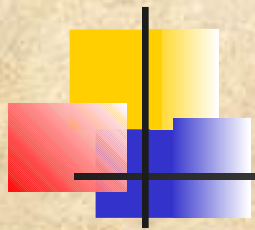


Chapter 7: TCP over Ad Hoc Networks

Table of

Contents

- Introduction
- TCP Protocol Overview
 - Designed and Fine-Tuned to Wired Networks
 - TCP Basics
 - TCP Header Format
 - Congestion Control
 - Round-Trip Time Estimation
- TCP and MANETs
 - Effects of Partitions on TCP
 - Impact of Lower Layers on TCP
- Solutions for TCP over Ad Hoc
 - Mobility-Related
 - Fairness-Related
- Conclusions and Future Directions



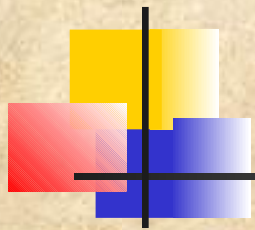
Introduction

- **TCP most widely used transport protocol**
- **Ad hoc networks composed exclusively of wireless links**
- **All nodes can move freely and unpredictably**
- **TCP needs to distinguish the nature of errors**



Designed and Fine Tuned to Wired Network

- Design heavily influenced by “end-to-end argument”
- Excessive intelligence in physical and link layers to handle error control, encryption or flow control
- Performance often dependant on flow control and congestion control
- Time outs and retransmission handle error control
- It is important to incorporate link layer acknowledgement and error detection/correction functionality
- High error rates, longer delays and mobility makes MANET environments extremely challenging



TCP Basics

- **Byte Stream Delivery:** TCP interfaces between the application layer above and the network layer below and TCP decides whether to segment or delineate the byte stream in order to transmit data in manageable pieces to the receiver, hence called “byte stream delivery service”
- **Connection-Oriented:** Two communicating TCP entities (the sender and the receiver) must first agree upon the willingness to communicate
- **Full-Duplex:** TCP almost always operates in full-duplex mode, and TCP exhibit asymmetric behavior only during connection start and close sequences (i.e., data transfer in the forward direction but not in the reverse, or vice versa)

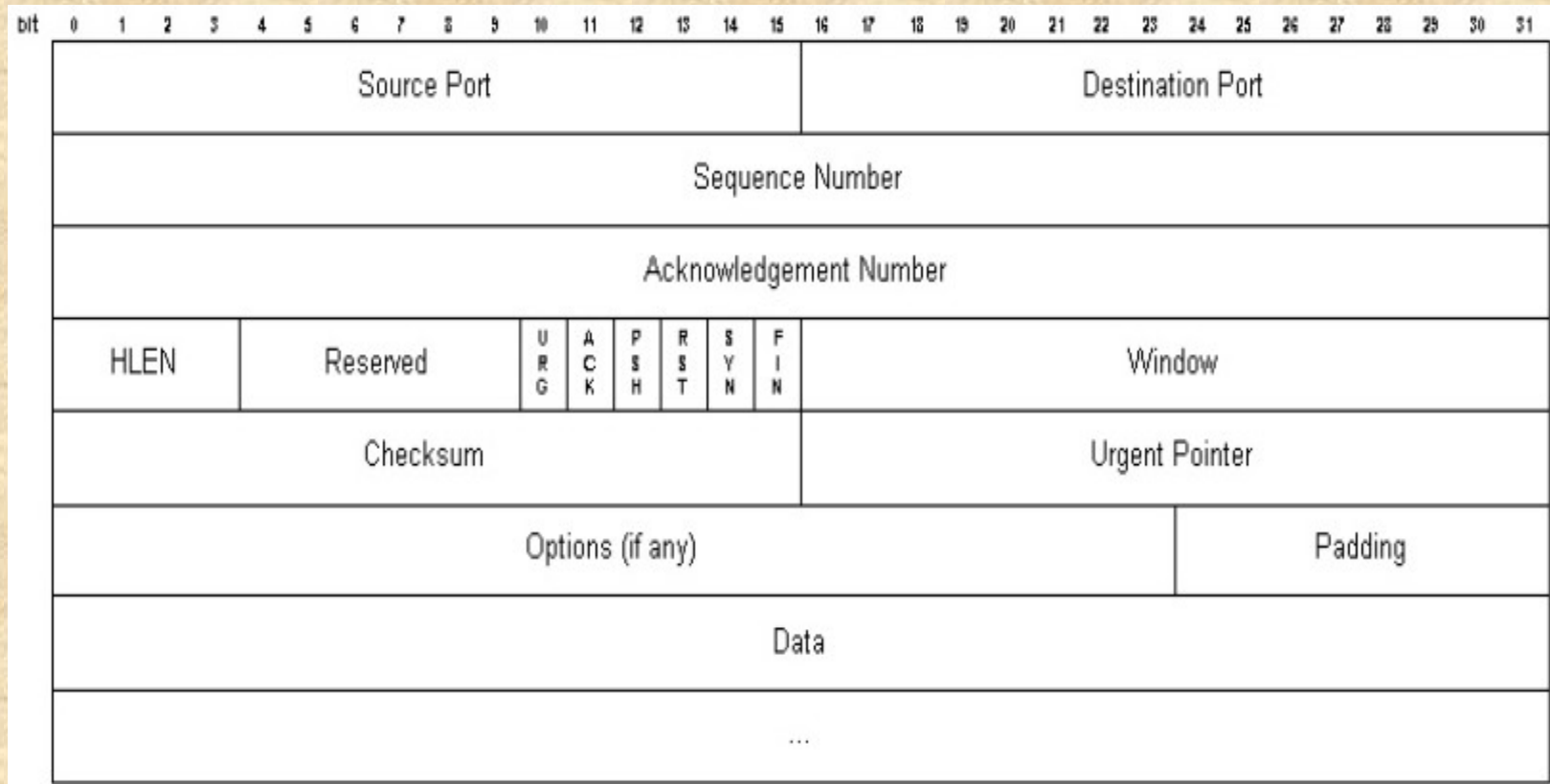


Reliable TCP guarantees

A number of mechanisms help provide the guarantees:

- **Checksums:** All TCP segments carry a checksum, used by the receiver to detect errors with either the TCP header or data
- **Duplicate data detection:** TCP keeps track of bytes received in order to discard duplicate copies of data that has already been received
- **Retransmissions:** TCP must implement retransmission schemes for data that may be lost or damaged and the lack of positive acknowledgements, coupled with a timeout period calls for a retransmission
- **Sequencing:** It is TCP's job to properly sequence segments it receives so that it can deliver the byte stream data to an application in order
- **Timers:** TCP maintains various static and dynamic timers on data sent and if the timer expires before receiving an acknowledgement, the sender can retransmit the segment

TCP Header Format





TCP Frame Details

- **Source Port:** This is a 16-bit number identifying the application where the TCP segment originated from within the sending host
- **Destination Port:** A 16-bit number identifying the application the TCP segment is destined for on a receiving host
- **Sequence Number:** A 32-bit number, identifying the current position of the first data byte in the segment and after reaching $2^{32} - 1$, this number will wrap around to 0
- **Acknowledgement Number:** A 32-bit number identifying the next data byte the sender expects from the receiver and is one greater than the most recently received data byte
- **Header Length:** A 4-bit field that specifies the total TCP header length in 32-bit words (or in multiples of 4 bytes), with the largest TCP header of 60 bytes
- **Reserved:** A 6-bit field currently unused and reserved for future use



Control Bits

- **Urgent Pointer (URG)** – If this bit field is set, the receiving TCP should interpret the urgent pointer field
- **Acknowledgement (ACK)** – If this bit is set, the acknowledgment field is valid
- **Push Function (PSH)** – If this bit is set, the receiver should deliver this segment to the receiving application as soon as possible
- **Reset Connection (RST)** – If this bit is present, it signals the receiver that the sender is aborting the connection and all the associated queued data and allocated buffers can be freely relinquished
- **Synchronize (SYN)** – When present, this bit field signifies that the sender is attempting to “synchronize” sequence numbers
- **No More Data from Sender (FIN)** – If set, this bit field tells the receiver that the sender has reached the end of its byte stream for the current TCP connection



TCP Details

- **Window:** This is a 16-bit integer used by TCP for flow control in the form of a data transmission window size
- **Checksum:** A sender computes the checksum value of 16-bits, based on the contents of the TCP header and data fields and is compared with the value the receiver generates using the same computation
- **Urgent Pointer:** This 16-bit field tells the receiver when the last byte of urgent data in the segment ends
- **Options:** Depending on the option(s) used, the length of this field varies in size, but it cannot be larger than 40 bytes due to the maximum size of the header length field (4 bits)
- **Padding:** It may be necessary to “pad” the TCP header with zeroes so that the segment ends on a 32-bit word boundary as defined by the standard
- **Data:** This variable length field carries the application data from TCP sender to receiver and this field coupled with the TCP header fields constitutes a TCP segment



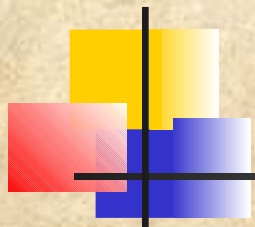
Congestion Control

- **Slow Start:** (A mechanism to control the transmission rate) Whenever a TCP connection starts, the Slow Start algorithm at the sender initializes a *congestion window* (CWND) to one segment and the congestion window increases by one segment for each acknowledgement returned
- **Congestion Avoidance:** When Slow Start forces a network to drop one or more packets due to overload or congestion, Congestion Avoidance is used to reduce the transmission rate
- **Fast Retransmit:** When a **duplicate ACK** is received, the sender does not know if this is because a TCP segment was lost or because a segment was delayed and received out of order at the receiver and if more than two duplicate ACKs are received by the sender, it does not even wait for the Retransmission Timeout to expire and retransmits the segment (as indicated by the position of the duplicate ACK in the byte stream)
- **Fast Recovery:** The sender has implicit knowledge that there is data still flowing to the receiver since duplicate ACKs can only be generated when a segment is received and the sender only enters Congestion Avoidance mode



Time Estimation

- **Round-Trip Time Estimation:** When a host transmits a TCP packet to its peer, and the reply does not come within the expected period, the packet is assumed to have been lost and the data is retransmitted
- **Over an Ethernet, no more than a few microseconds should be needed for a reply**
- **This process called Round-Trip Time (RTT) estimation**
- **If the RTT estimate is too low, packets are retransmitted unnecessarily; if too high, the connection can sit idle while the host waits to timeout**



TCP and Manets

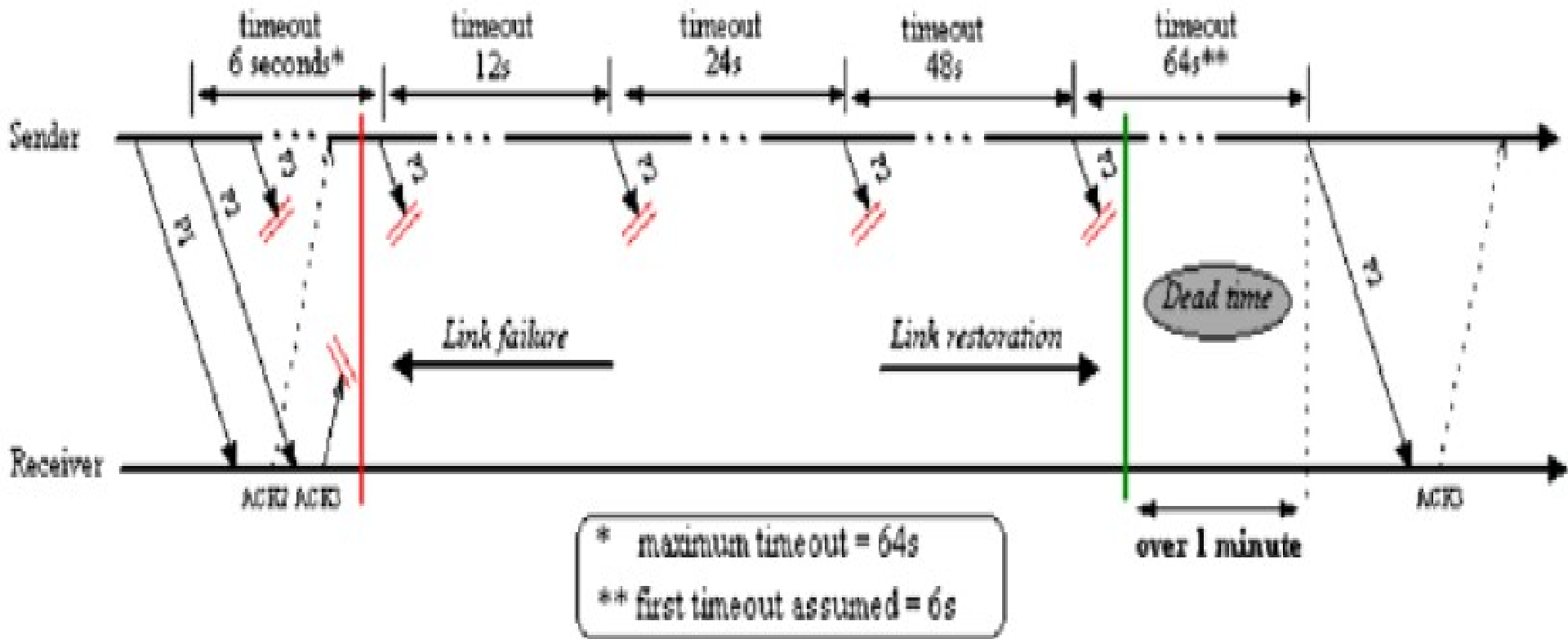
Challenges

- As the topology changes, the path is interrupted and TCP goes into repeated, exponentially increasing time-outs, with severe performance impact
- TCP performance in ad hoc multi-hop environment depends critically on the congestion window in use

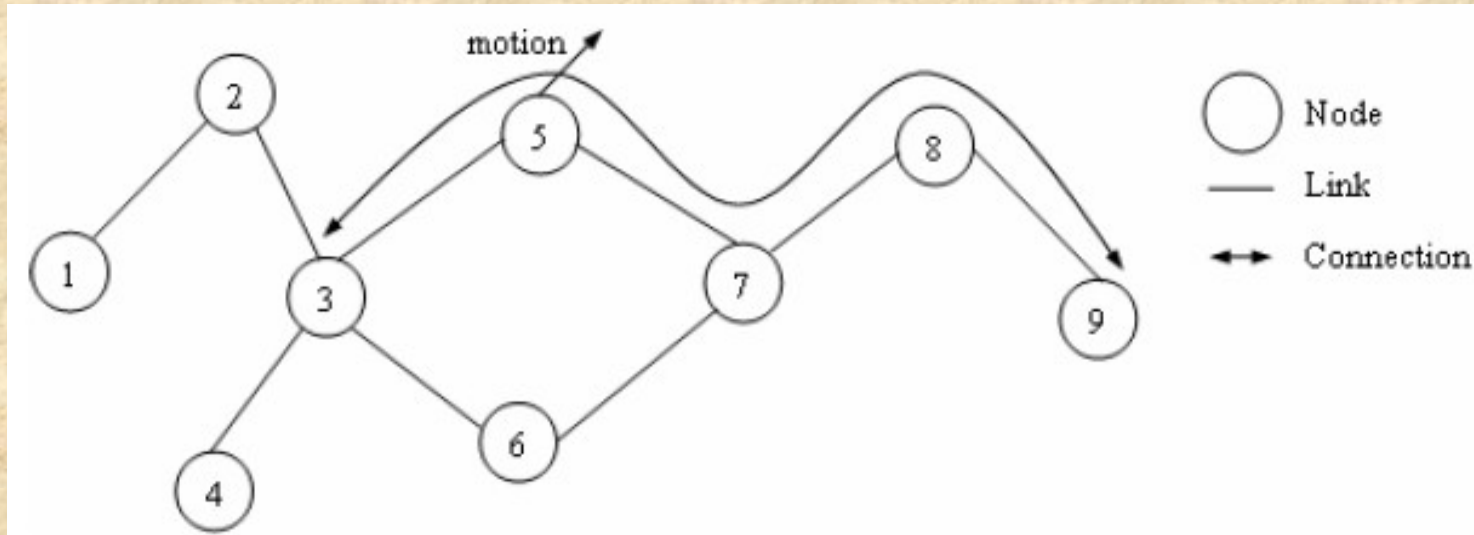
Significant TCP unfairness

- TCP injects packets at an increasing rate into the network until a packet loss is detected and then, the sender shrinks its CWND, retransmits the lost packet and resumes transmission at a lower increasing rate
- If the losses persist at every retransmission, the sender doubles its wait timer (i.e., the RTO) so that it can wait longer for the ACK of the current packet being transmitted and is known as the *exponential back off strategy*

Drawback of TCP Exponential Back Off

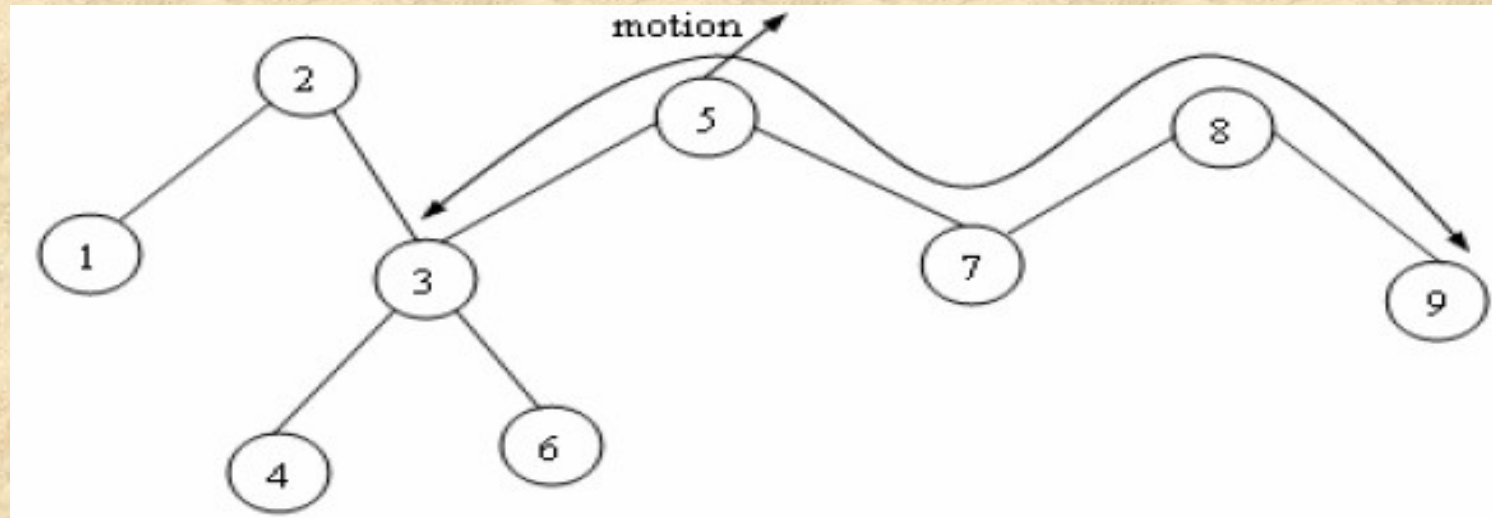


Effects of Partitions on TCP



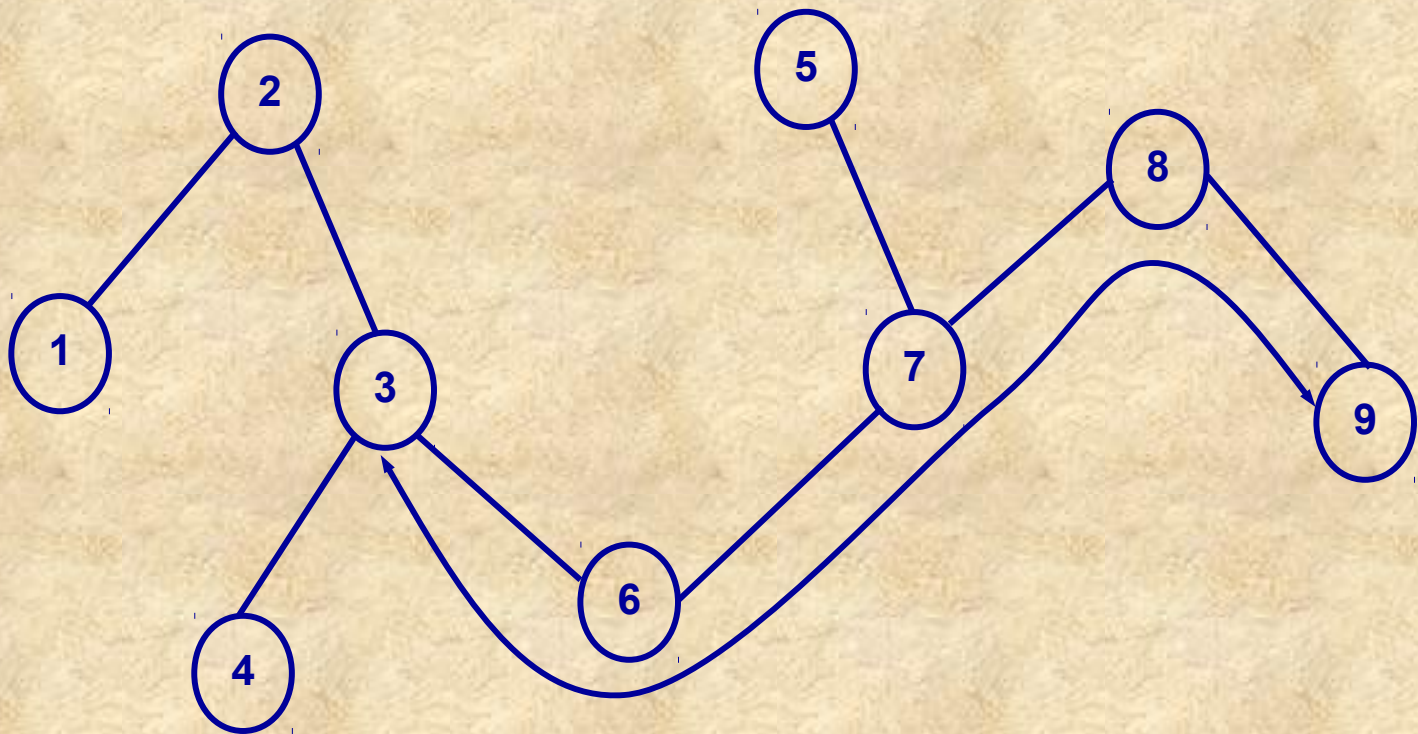
Node 5 moves away from node 3 (short-term partition)

Long Term Partition

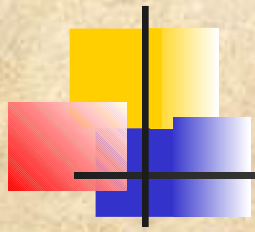


Node 5 moves away from node 3 (long-term partition)

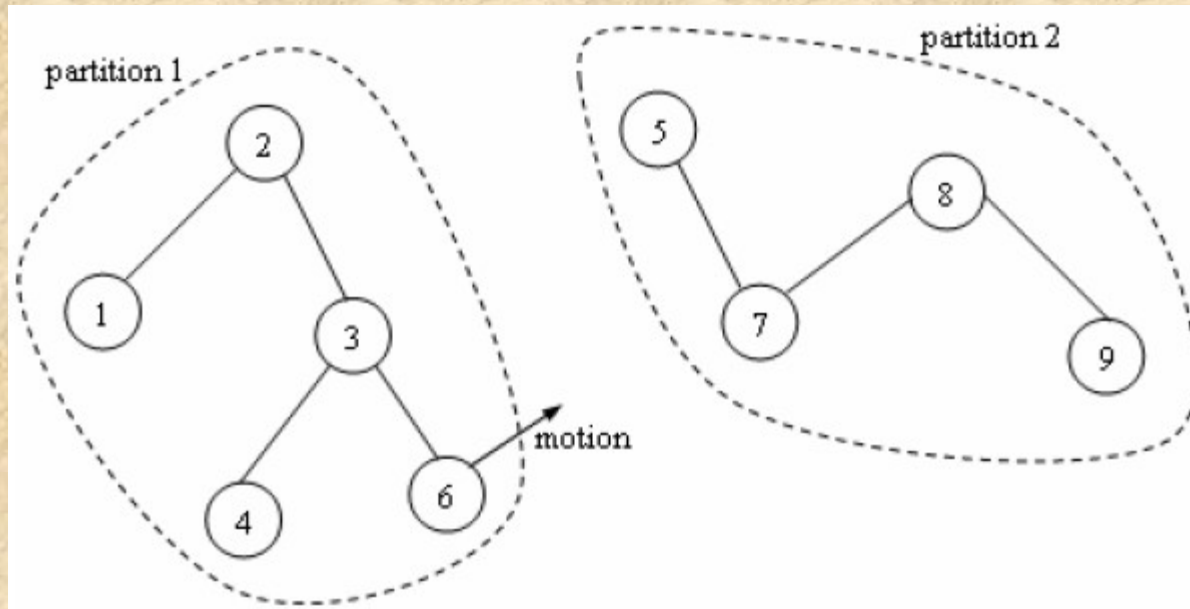
Reestablishing Path



The routing protocol reestablishes the path through node 6



Long Term Network Partition



No communication between the partitions



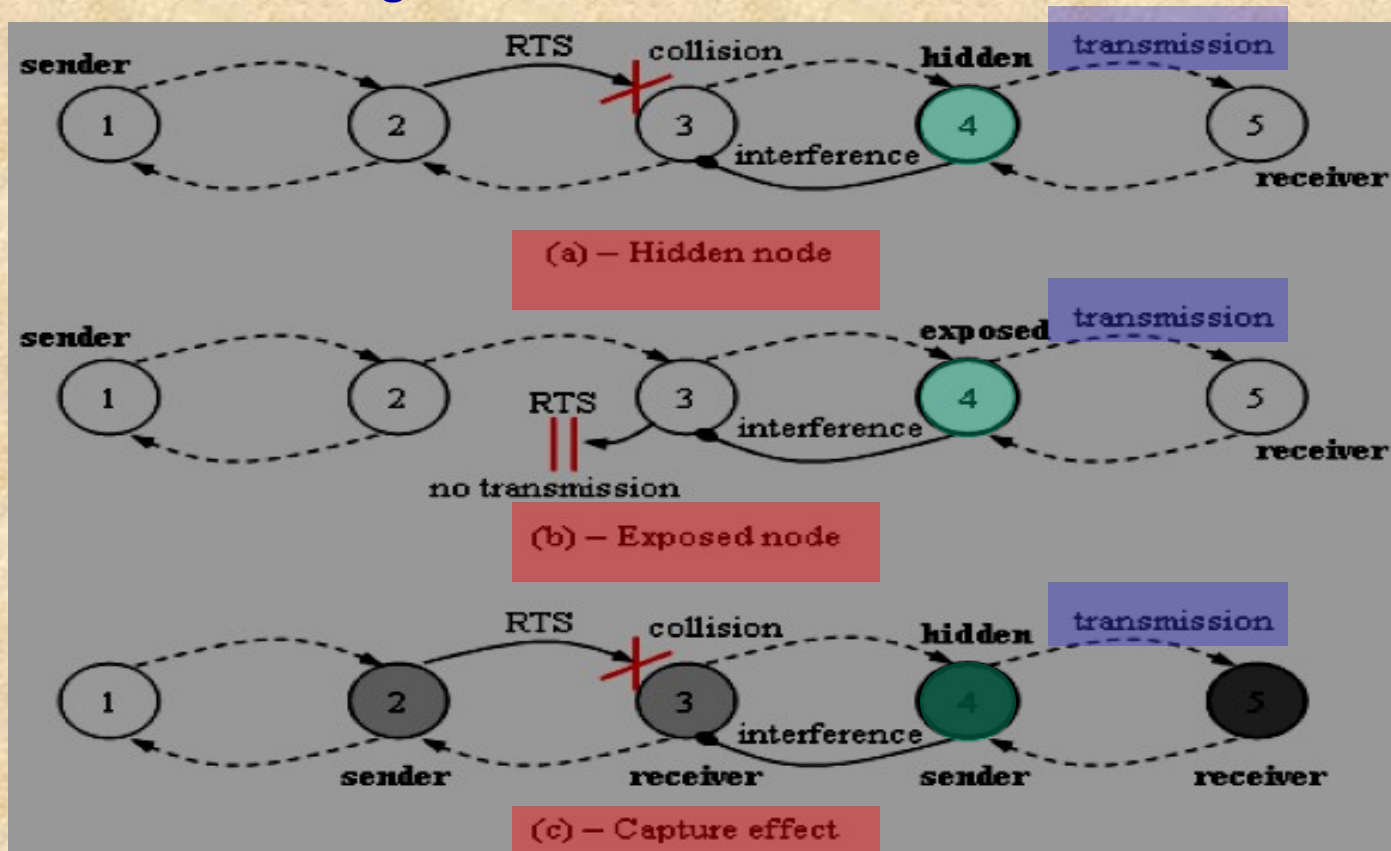
Impact of Lower Layers on TCP

MAC Layer Impact:

- It is intended for providing an efficient shared broadcast channel through which the involved mobile nodes can communicate
 - In IEEE 802.11, RTS/CTS handshake is only employed when the DATA packet size exceeds some predefined threshold
 - Each of these frames carries the remaining duration of time for the transmission completion, so that other nodes in the vicinity can hear it and postpone their transmissions
 - The nodes must await an IFS interval and then contend for the medium again
 - The contention is carried out by means of a binary exponential backoff mechanism which imposes a further random interval
 - At every unsuccessful attempt, this random interval tends to become higher

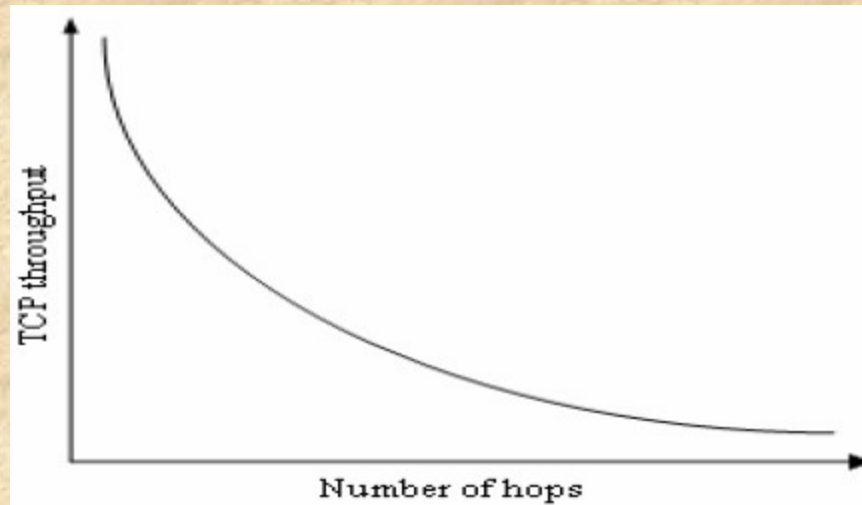
Issues at the MAC Layer

- Consider a linear topology in which each node can only communicate with its adjacent neighbors
- In addition, consider that in Figures 7.7(a) and 7.7(b) there exist a single TCP connection running between nodes 1 and 5



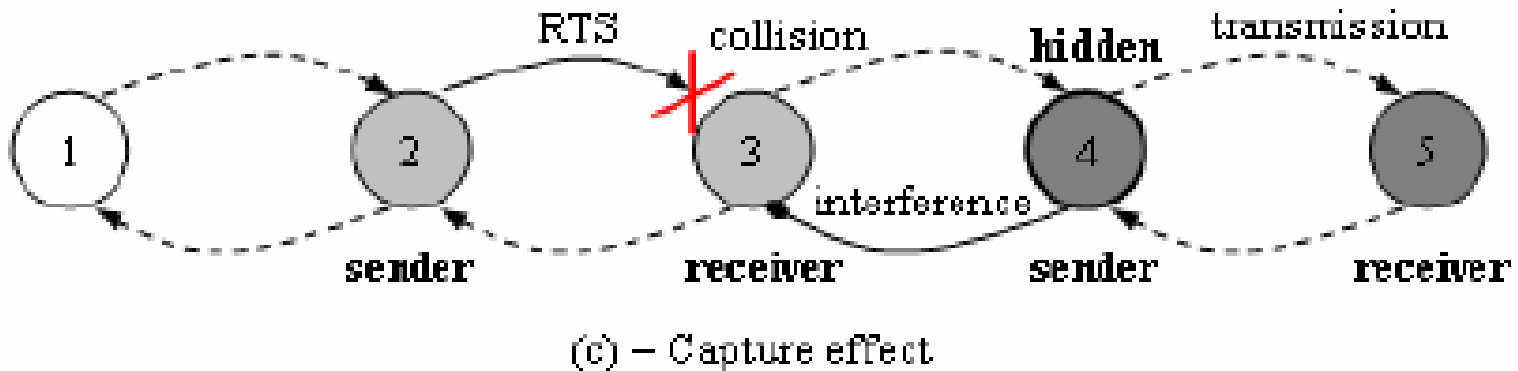
TCP Throughput

Larger the number of nodes a TCP connection needs to span, lower is the end-to-end throughput, as there will be more medium contention taking place in several regions of the network

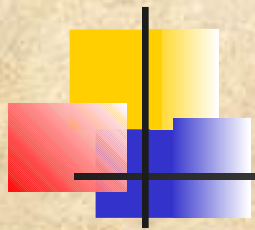


TCP throughput is inversely proportional to the number of hops

Capture Conditions



- In Figure 7.7(c) where there are two independent connections, (connection 2-3) (connection 4-5)
- Assuming that connection 2-3 experiences collision due to the hidden node problem caused by the active connection 4-5, node 2 will back off and retransmit the lost frame
- At every retransmission, the binary exponential backoff mechanism imposes an increasingly backoff interval, and implicitly, this is actually decreasing the possibility of success for the connection 2-3 to send a packet as connection 4-5 will “dominate” the medium access once it has lower backoff value
- In consequence, the connection 2-3 will hardly obtain access to the medium while connection 4-5 will capture it



Network Layer Impact

- **Routing strategies play a key role on TCP performance**
- **There have been a lot of proposed routing schemes and, typically, each of them have different effects on the TCP performance**

- **DSR protocol operates on an on-demand basis in which a node wishing to find a new route broadcasts a RREQ packet**
- **The problem with this approach concerns the high probability of stale routes in environments where high mobility as well as medium constraints may be normally present**
- **The problem is exacerbated by the fact that other nodes can overhear the invalid route reply and populate their buffers with stale route information**
- **It can be mitigated by either manipulating TCP to tolerate such a delay or by making the delay shorter so that the TCP can deal with them smoothly**



TORA

- TORA has been designed to be highly dynamic by establishing routes quickly and concentrating control messages within a small set of nodes close to the place where the topological change has occurred
- TORA makes use of directed acyclic graphs, where every node has a path to a given destination and established initially
- This protocol can also suffer from stale route problem similar to the DSR protocol
- The problem occurs mainly because TORA does not prioritize shorter paths, which can yield considerable amount of out-of-sequence packets for the TCP receiver, triggering retransmission of packets



Path Asymmetry Impact

- **In ad hoc networks, asymmetry can occur by different reasons including lower layer strategies**
- **Loss Rate Asymmetry:** It takes place when the backward path is significantly more error prone than the forward path
- **Bandwidth Asymmetry:** Here, forward and backward data follow distinct paths with different speeds and this can happen in ad hoc networks as well, since all nodes need not have the same interface speed
- **Media Access Asymmetry:** This type of asymmetry may occur due to characteristics of the wireless shared medium as TCP ACKs may have to contend for the medium along with TCP data, which may cause excessive delay as well as drops of TCP ACK packets



Route Assymetry

- Route asymmetry implies in distinct paths in both directions
- Route asymmetry is associated with the possibility of different transmission ranges for the nodes
- The inconvenience with different transmission ranges is that it can lead to conditions in which the forward data follow a considerably shorter path than the backward data (TCP ACK) due to lack of power in one (or more) of the nodes in the backward path
- However, multi-hop paths are prone to have low throughput and TCP ACKs may face considerable disruptions



Solutions for TCP over Ad Hoc

- Mobility-Related

- TCP-Feedback

- TCP sender can effectively distinguish between route failure and network congestion by receiving Route Failure Notification (RFN) messages from intermediate nodes
- Upon receipt of a Route Re-establishment Notification (RRN) message from the routing protocol, the sender leaves the frozen state and resumes transmission using the same variables values prior to the interruption
- A route failure timer is employed to prevent infinite wait for RRN messages, is started whenever a RFN is received and upon expiration of this timer, the frozen timers of TCP are reset hence allowing the TCP congestion control to be invoked normally



The ELFN Approach

- **Explicit Link Failure Notification (ELFN) is a cross-layer proposal in which TCP also interacts with the routing protocol in order to detect route failure and take appropriate actions**
- **ELFN messages are sent back to the TCP sender from the node detecting the failure**
- **ELFN messages contain sender and receiver addresses and ports, as well as the TCP sequence number**
- **Whenever the TCP sender receives an ELFN message, it enters a “stand-by” mode in which its timers are disabled and probe packets are sent regularly towards the destination in order to detect route restoration**
- **Upon receiving an ACK packet, the sender leaves the “stand-by” mode and resumes transmission using its previous timer values and state variables**



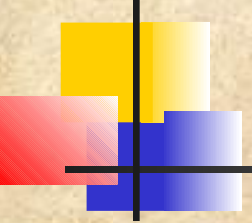
Fixed Retransmission Timeout (RTO)

- Relies on the idea that routing error recovery should be accomplished in a fast fashion by the routing algorithm
- It disables such a mechanism whenever two successive retransmissions due to timeout occur, assuming that it actually indicates route failure
- TCP sender doubles the RTO once and if the missing packet does not arrive before the second RTO expires, the packet is retransmitted again and again, but the RTO is no longer increased



The ATCP Protocol

- The Ad hoc TCP (ATCP) protocol does not impose changes to the standard TCP itself and instead, it implements an intermediate layer between the network and the transport layers in order to provide an enhanced performance to TCP
- ATCP relies on the ICMP protocol and on the Explicit Congestion Notification (ECN) scheme to detect / distinguish network partition and congestion, respectively
- The intermediate layer keeps track of the packets to and from the transport layer so that the TCP congestion control is not invoked when it is not really needed



TCP-DOOR (Detection Out-Of-Order and Response)

- Mobility in MANETs is extremely frequent and the packet usually arrive out-of-order (OOO) at the destination
- The TCP-DOOR (Detection of Out-Of-Order and Response) protocol focuses on the idea that OOO delivery of packets can happen frequently in MANETs as a result of nodes mobility
- TCP-DOOR implements a detection of such deliveries at both entities: TCP sender and TCP receiver



Main Drawbacks

- The approaches that rely on feedback information from inside the network (TCP-F, ELFN-based, ATCP) may fail in situations where TCP sender is unable to receive data from the next hop node
- The usage of explicit notification by the intermediate nodes, such as ECN, raises many security concerns
- The assumption in TCP-DOOR that OOO packets are exclusive results of route disturbance may not be true in a quite a few scenarios
- The main concern addressed by the approaches presented so far is how to avoid the TCP exponential backoff mechanism when **losses** take place by **factors** other than congestion

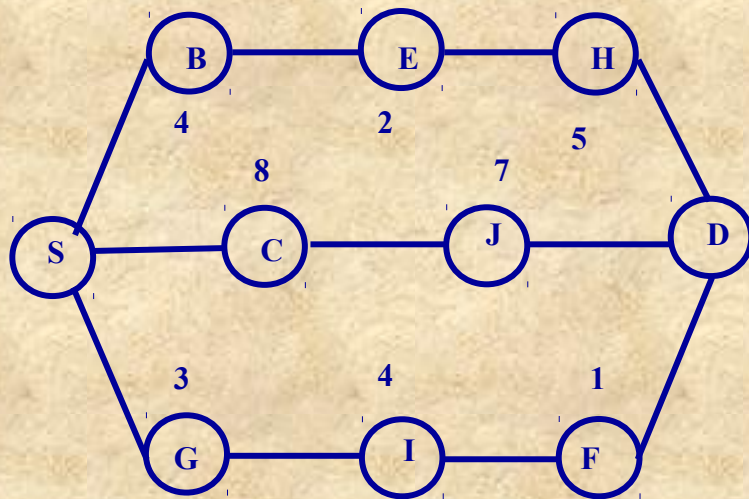


Fairness- Related

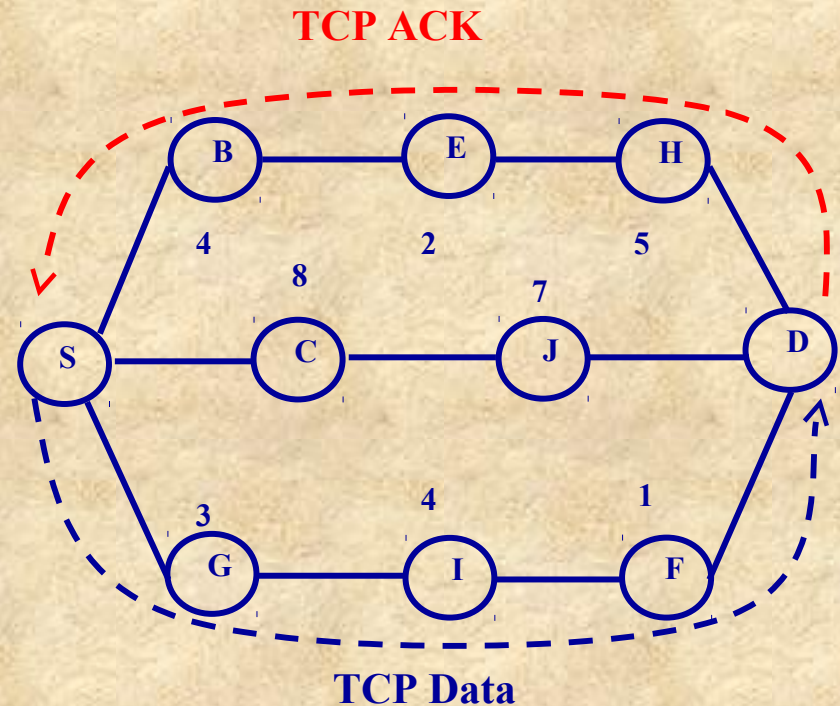
COPAS

- A protocol called COPAS (COn contention-based PAtH Selection) has been proposed to address TCP performance drop due to the capture problem and resulting unfairness
- COPAS implements two novel routing techniques in order to contention-balance the network, namely, the use of disjoint forward (for TCP data) and reverse (for TCP ACK) paths to reduce the conflicts between TCP packets traveling in opposite directions

Route Establishment in COPAS

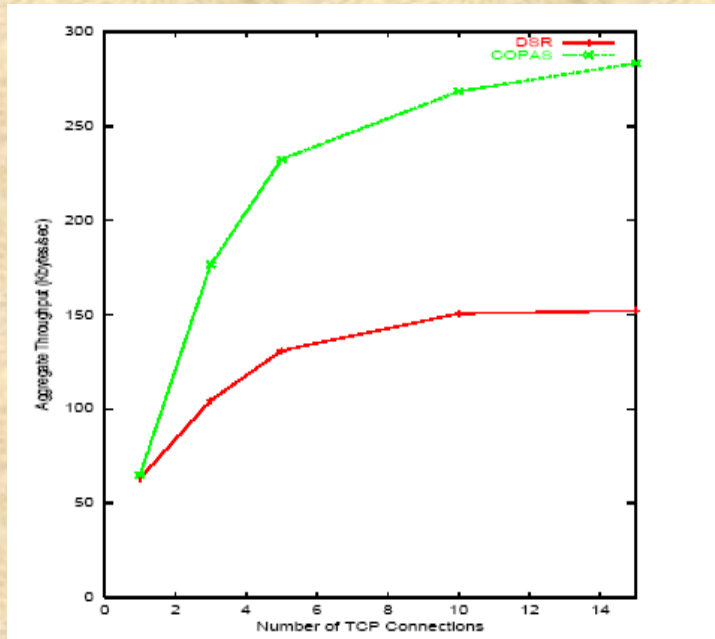


(a) – Network contention perceived at node D

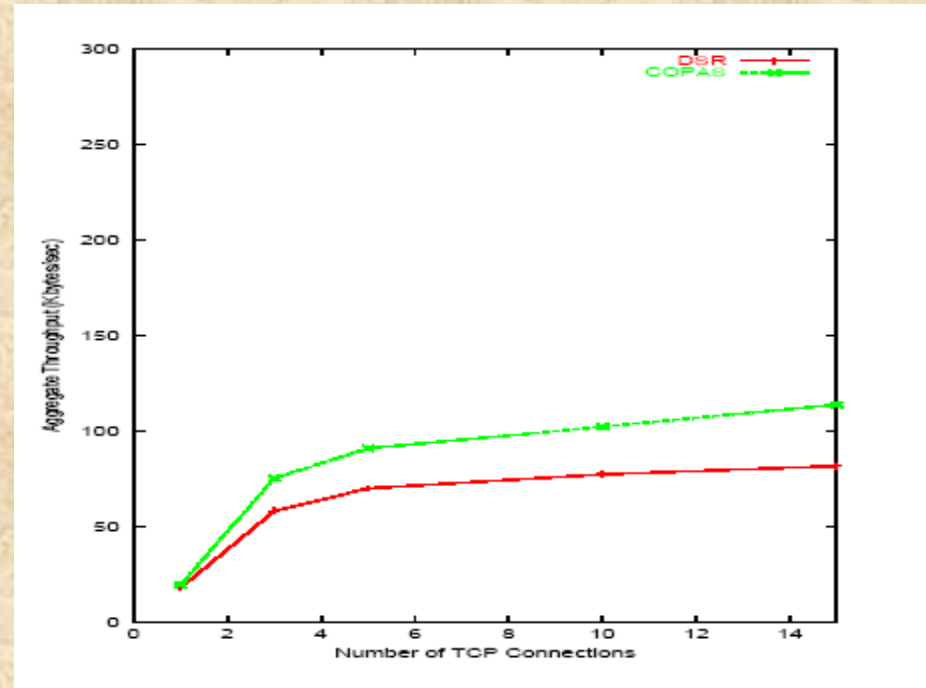


(b) – Routes selected by node D

Average Aggregate Throughput

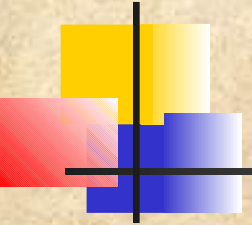


50 Nodes



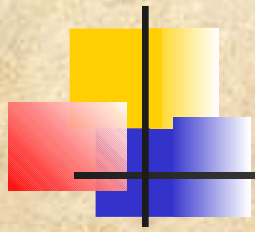
100 Nodes

Simulation results of COPAS applied to scenario of 50 and 100 nodes

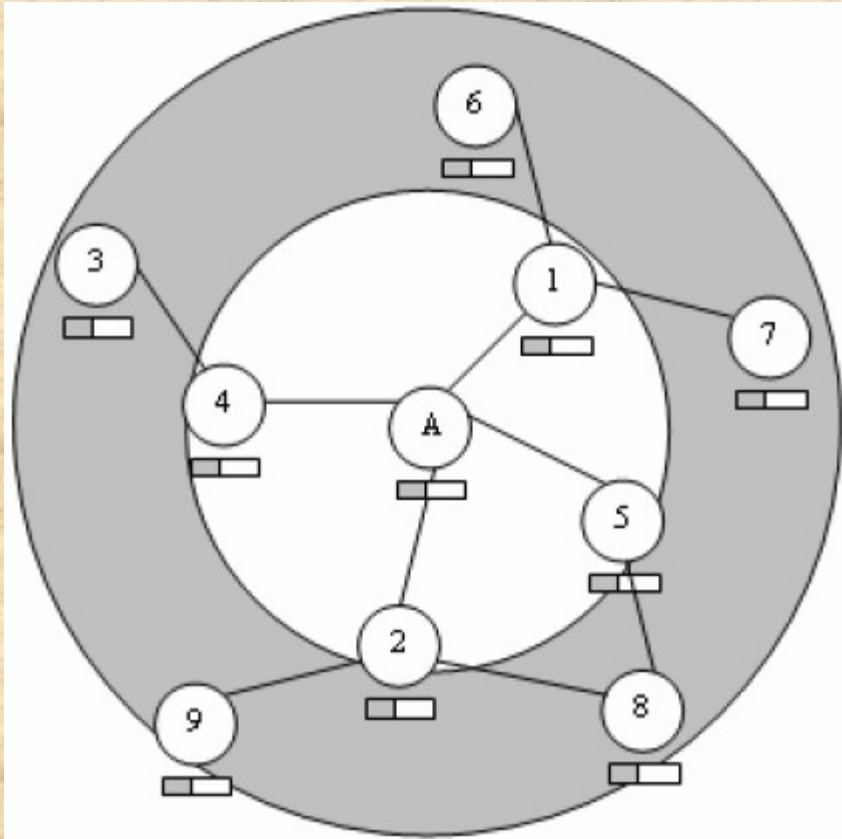


Neighborhood RED (Randomly Early Detection)

- Two unique features of ad hoc wireless networks are the key to understand unfair TCP behaviors: Spatial reuse constraint and the location dependency
- View a node and its interfering neighbors to form a neighborhood (the neighborhood of a node X is formed by all nodes within communication range of X)
- Flows get different feedback in terms of packet loss rate and packet delay when congestion happens
- The main achievement of NRED is the ability to detect early congestion and drop packets proportionally to a flow's channel bandwidth utilization



Node A's Neighborhood and Distributed Queue



- Keep estimating the size of neighborhood queue
- Once queue size exceeds certain threshold, a drop probability is computed
- This is propagated to provide cooperative packet drop



Conclusions and Future Directions

- Concerning the error-detection strategies used in each approach, they may be classified as network detection and end node detection
- Each approach has its advantages and disadvantages, and ideally, it is better to combine the advantages of each one
- The interactions between TCP and MAC protocols could be improved by using either using smaller values for the maximum TCP window size or larger MAC IFS intervals, respectively
- It might be useful to investigate the possibility of increasing the maximum number of possible retransmissions at the MAC layer as an attempt to increase the probability of success of the local retransmission scheme
- With regards to multipath routing strategies, further evaluation towards improvements with respect to TCP support is needed
- Power management is a very important topic within MANETs, as they are supposed to be composed mostly of battery powered devices
- Thus, power aware approaches offer increasing interest while little has been done with regards to TCP, as is not power aware
- Interoperation between wireless mobile ad hoc networks and wired networks is another subject that has not been adequately addressed from TCP perspective
- Security considerations have become nowadays a hot issue in wireless environments as wireless mediums are much more susceptible to malicious users than the wired ones